IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants:	Srinivasamurthy et al.	Patent Application	
Application No.:	10/630,913	Group Art Unit:	2192
Filed:	July 31, 2003	Examiner:	Dao, T.

For: APPLICATION SPECIFIC OPTIMIZATION OF INTERPRETERS FOR EMBEDDED SYSTEMS

REPLY BRIEF

In response to the Examiner's Answer mailed on May 12, 2010, Appellants respectfully submit the following remarks.

REMARKS

Appellants submit the following remarks in response to the Examiner's Answer. In these remarks, Appellants address certain arguments presented in the Examiner's Answer. While only certain arguments are addressed in this Reply Brief, this should not be construed that Appellants agree with the other arguments presented in the Examiner's Answer.

1. Response to the Obviousness Arguments on Pages 13-16 of Examiner's Answer

The Examiner's Answer states the following:

> That is to say, what so-called and argued for "statically" process, here, is where the "compiled Java application" is parsed (bytecode stream 504 in Sokolov's FIG. 5 is read/parsed) for detecting/selecting and embedding "the semantically enriched opcodes" (selecting and embedding 508 into 510, also in Sokolov's FIG. 5) to optimize execution of the interpreter-based runtime system, i.e., that is to say, before runtime execution in a Java Virtual Machine ("statically embedding" before runtime execution-emphasis added).

(Emphasis added; Examiner's Answer, page 14, first paragraph.) and

> Sokolov teaches Java Bytecode Verifier 500 reads/parses the input stream 504 embeds the macro instruction 508 ("semantically enriched opcodes") in a (single) verification phase (col.6: 46-47 and 63-65), and all said steps in the (single) verification phase are performed before runtime execution by a JVM (col.3: 42-48 and col. 7: 50-59) – emphasis added. thus, the step of embedding the macro instruction 508 into the reduced and/or optimized stream 510 is indeed a so-called "statically embedding" process and not a dynamic process as Appellants characterized of Sokolov teachings.

(Emphasis added; Examiner's Answer, page 16, second paragraph.)

Appellants respectfully disagree with the present conclusions of the Examiner's Answer. For example, Appellants respectfully note that the instant Application recites, in regards to the "statically embedding the semantically enriched opcode to optimize execution of the interpreter-based runtime system" of Appellants' Claim 1, the following:

> In the static sEc detection phase, the particular compiled Java application is parsed for the most repetitive longest sequence of Java bytecodes. sEc bytecodes are selected from these based on cost and control flow criteria. This approach does not capture the dynamic semantics of the application as the bytecode patterns may not account for the dominant dynamic behaviour of the Java application. Thus, this detection technique is better suited for space optimization of Java applications on an embedded platform. In effect, this technique captures spatial localization of the Java bytecode rather than temporal localization.

(Appellants' specification, page 9, lines 14-20.) In contrast, Sokolov et al. (U.S. Patent Application No. 6,988,261) (hereinafter, "Sokolov") states the following:

> As will be appreciated, the Java Bytecode verifier 500 can identify a number of predetermined sequences of Java Bytecode instructions and replace them with the appropriate Java macro instruction. The Java Bytecode verifier 500 can also be implemented to analyze the sequences that appear in the stream 504 and replace only those that meet a criteria (e.g., a sequence that has appeared more than a predetermined number of times). In any case, the number of Java Bytecode instructions in an input stream 504 (e.g., stream 504) can be reduced significantly. Thus, the performance of virtual machines, especially those operating with limited resources, can be enhanced.

(Emphasis added; Sokolov, column 7, lines 47-59.) Appellants respectfully submit that Sokolov describes a process that would need to be dynamically repeated each time an application is run and is "temporal" in nature, which appears to Appellants to be the opposite of Appellants' technique which captures "spatial localization of the Java bytecode", as described in the specification (see above), which is also further descriptive of the words in Claim 1. Therefore,

Appellants respectfully submit that Sokolov teaches away from Claim 1, as submitted in the Appeal Brief.

Therefore, Appellants respectfully submit that Appellant's Claim 1 overcomes the 35 U.S.C. §103(a) rejections and is in condition for allowance. Furthermore, Appellants respectfully submit that Claim 30 is in condition for allowance for the reasons stated herein with regards to Claim 1. Moreover, Appellants respectfully submit that Claims 21-29 depending on Claim 1 and Claims 31-39 depending on Claim 30 are in condition for allowance as being dependent on an allowable base claim.

CONCLUSION

In view of the above remarks, Appellants continue to assert that Claims 1 and 21-39 are directed toward patentable subject matter. As such, Appellants respectfully request that the rejections of Claims 1 and 21-39 be reversed.

The Appellants wish to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellants' undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,

WAGNER BLECHER

Dated: 07/11/2010

/John P. Wagner, Jr./
John P. Wagner, Jr.
Registration Number: 35,398

WAGNER BLECHER
123 Westridge Drive
Watsonville, CA 95076
(408) 377-0500